WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Friday, September 10, 2004

Hide?	Set Name	Query	Hit Count
	DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ		
□	L8	L7 and L1	23
□	L7	L4	149
	DB=EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
□	L6	L3	2
□	L5	L4	0
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
□	L4	L3 and pars$	149
□	L3	L2 and stub	393
□	L2	instrument$ and compil$	13996
	DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ		
□	L1	717/124-135,140-144.ccls.	2355

END OF SEARCH HISTORY

Terms used **instrument compile parse stub**

Found **36** of **142,346**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 36

Result page: **1** 2 next

Relevance scale

1 A tool framework for static and dynamic analysis of object-oriented software with templates
Kathleen A. Lindlan, Janice Cuny, Allen D. Malony, Sameer Shende, Forschungszentrum Juelich, Reid Rivenburgh, Craig Rasmussen, Bernd Mohr
November 2000 **Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM)**
Full text available: pdf(2.67 MB) Publisher Site
Additional Information: full citation, abstract, references, citings, index terms

The developers of high-performance scientific applications often work in complex computing environments that place heavy demands on program analysis tools. The developers need tools that interoperate, are portable across machine architectures, and provide source-level feedback. In this paper, we describe a tool framework, the Program Database Toolkit (PDT), that supports the development of program analysis tools meeting these requirements. PDT uses compile-time information to create a comp ...

2 Fast detection of communication patterns in distributed executions
Thomas Kunz, Michiel F. H. Seuren
November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**
Full text available: pdf(4.21 MB)
Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

3 Parse tree annotations
James J. Purtilo, John R. Callahan
December 1989 **Communications of the ACM**, Volume 32 Issue 12
Full text available: pdf(1.09 MB)
Additional Information: full citation, abstract, references, citings, index terms, review

A technique for associating rewrite rules with productions so that many high-level transformations of a source file can be generated easily is described. While eclipsed in power by other editing and compiler generation systems supporting management of both synthesized and inherited attributes, this approach is especially simple to employ and is sufficient in power to deal with a wide class of problems arising from practical applications.

Keywords: Attribute grammars, parsers, software interconnection systems, source-to-source transformations

4 Flick: a flexible, optimizing IDL compiler
Eric Eide, Kevin Frei, Bryan Ford, Jay Lepreau, Gary Lindstrom
May 1997 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1997 conference on Programming language design and implementation**, Volume 32 Issue 5
Full text available: pdf(1.75 MB) Additional Information: full citation, abstract, references, citings, index terms

An interface definition language (IDL) is a nontraditional language for describing interfaces between software components. IDL compilers generate "stubs" that provide separate communicating processes with the abstraction of local object invocation or procedure call. High-quality stub generation is essential for applications to benefit from component-based designs, whether the components reside on a single computer or on multiple networked hosts. Typical IDL compilers, ...

5 A high performance Erlang system
Erik Johansson, Mikael Pettersson, Konstantinos Sagonas
September 2000 **Proceedings of the 2nd ACM SIGPLAN international conference on Principles and practice of declarative programming**
Full text available: pdf(320.62 KB) Additional Information: full citation, references, citings, index terms

6 Middleware performance analysis: Performance monitoring of java applications
M. Harkema, D. Quartel, B. M. M. Gijsen, R. D. van der Mei
July 2002 **Proceedings of the third international workshop on Software and performance**
Full text available: pdf(219.69 KB) Additional Information: full citation, abstract, references, citings, index terms

Over the past few years, Java has evolved into a mature platform for developing enterprise applications. A critical factor for the commercial success of these applications is end-to-end performance, e.g., in terms of response times, throughput and availability. This raises the need for the development, validation and analysis of performance models to predict performance metrics of interest. To develop and validate performance models, insight in the execution behavior of the application is essent ...

Keywords: performance measurement and monitoring of java applications

7 The architecture of Montana: an open and extensible programming environment with an incremental C++ compiler
Michael Karasick
November 1998 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 6th ACM SIGSOFT international symposium on Foundations of software engineering**, Volume 23 Issue 6
Full text available: pdf(1.16 MB) Additional Information: full citation, abstract, references, citings, index terms

Montana is an open, extensible integrated programming environment for C++ that supports incremental compilation and linking, a persistent code cache called a CodeStore, and a set of programming interfaces to the CodeStore for tool writers. CodeStore serves as a central source of information for compiling, browsing, and debugging. CodeStore contains information about both the static and dynamic structure of the compiled program. This information spans files, macros, declarations, function bodies, ...

Keywords: C++, conpilation, extensible systems, frameworks, incremental compilation, incremental development environments, programming environments

8 Viewing a programming environment as a single tool
Norman M. Delisle, David E. Menicosy, Mayer D. Schwartz
April 1984 **Proceedings of the first ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments**, Volume 19 , 9 Issue 5 , 3
Full text available: pdf(889.57 KB) Additional Information: full citation, abstract, references, citings, index terms

Programming environments support the creation, modification, execution and debugging of programs. The goal of integrating a programming environment is more than simply building tools that share a common data base and provide a consistent user interface. Ideally, the programming environment appears to the programmer as a single tool; there are no firewalls separating the various functions provided by the environment. This paper describes the techniques used to integrate Magpie, an interactiv ...

9 Towards automatic construction of staged compilers
Matthai Philipose, Craig Chambers, Susan J. Eggers
January 2002 **ACM SIGPLAN Notices , Proceedings of the 29th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**, Volume 37 Issue 1
Full text available: pdf(269.51 KB) Additional Information: full citation, abstract, references, citings

Some compilation systems, such as offline partial evaluators and selective dynamic compilation systems, support staged optimizations. A staged optimization is one where a logically single optimization is broken up into stages, with the early stage(s) performing preplanning set-up work, given any available partial knowledge about the program to be compiled, and the final stage completing the optimization. The final stage can be much faster than the original optimization by having much of its work ...

10 DDD papers: XAspects: an extensible system for domain-specific aspect languages
Macneil Shonle, Karl Lieberherr, Ankit Shah
October 2003 **Companion of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**
Full text available: pdf(218.84 KB) Additional Information: full citation, abstract, references, citings, index terms

Current general aspect-oriented programming solutions fall short of helping the problem of separation of concerns for several concern domains. Because of this limitation good solutions for these concern domains do not get used and the opportunity to benefit from separation of these concerns is missed. By using XAspects, a plug-in mechanism for domain-specific aspect languages, separation of concerns can be achieved at a level beyond what is possible for object-oriented programming languages. As ...

Keywords: aspect-oriented programming, domain-specific languages, generative programming, language extensions

11 Human-computer interface development: concepts and systems for its management
H. Rex Hartson, Deborah Hix
March 1989 **ACM Computing Surveys (CSUR)**, Volume 21 Issue 1
Full text available: pdf(7.97 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Human-computer interface management, from a computer science viewpoint, focuses on the process of developing quality human-computer interfaces, including their representation, design, implementation, execution, evaluation, and maintenance. This survey presents important concepts of interface management: dialogue independence, structural modeling, representation, interactive tools, rapid prototyping, development methodologies, and control structures. *Dialogue independence* is th ...

12 Reconciling responsiveness with performance in pure object-oriented languages
Urs Hölzle, David Ungar
July 1996 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 18 Issue 4

Full text available: pdf(537.19 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Dynamically dispatched calls often limit the performance of object-oriented programs, since opject-oriented programming encourages factoring code into small, reusable units, thereby increasing the frequency of these expensive operations. Frequent calls not only slow down execution with the dispatch overhead per se, but more importantly they hinder optimization by limiting the range and effectiveness of standard global optimizations. In particular, dynamically dispatched calles prevent stand ...

Keywords: adaptive optimization, pause clustering, profile-based optimization, run-time compilation, type feedback

13 Software reuse
Charles W. Krueger
June 1992 **ACM Computing Surveys (CSUR)**, Volume 24 Issue 2

Full text available: pdf(4.96 MB) Additional Information: full citation, abstract, references, citings, index terms

Software reuse is the process of creating software systems from existing software rather than building software systems from scratch. This simple yet powerful vision was introduced in 1968. Software reuse has, however, failed to become a standard software engineering practice. In an attempt to understand why, researchers have renewed their interest in software reuse and in the obstacles to implementing it. This paper surveys the different approaches to software reuse found in the ...

Keywords: abstraction, cognitive distance, software reuse

14 Adventures in interoperability: the SML.NET experience
Nick Benton, Andrew Kennedy, Claudio V. Russo
August 2004 **Proceedings of the 6th ACM SIGPLAN international conference on Principles and practice of declarative programming**

Full text available: pdf(434.04 KB) Additional Information: full citation, abstract, references, index terms

SML.NET is a compiler for Standard ML that targets the Common Language Runtime and is integrated into the Visual Studio development environment. It supports easy interoperability with other .NET languages via a number of language extensions, which go considerably beyond those of our earlier compiler, MLj.This paper describes the new language extensions and the features of the Visual Studio plugin, including syntax highlighting, Intellisense, continuous type inference and debugger support. We dis ...

Keywords: applications of declarative programming, functional programming, integration of paradigms, programming environments

15 Requirements for and evaluation of RMI protocols for scientific computing
Madhusudhan Govindaraju, Aleksander Slominski, Venkatesh Choppella, Randall Bramley, Dennis Gannon
November 2000 **Proceedings of the 2000 ACM/IEEE conference on Supercomputing (CDROM)**

Full text available: pdf(306.83 KB) Additional Information: full citation, abstract, references, citings, index

Publisher Site terms

Distributed software component architectures provide a promising approach to the problem of building large scale, scientific Grid applications. Communication in these component architectures is based on Remote Method Invocation (RMI) protocols that allow one software component to invoke the functionality of another. Examples include Java remote method invocation (Java RMI) and the new Simple Object Access Protocol (SOAP). SOAP has the advantage that many programming languages and component ...

Keywords: Distributed computing, software component systems, communicationprotocols, RMI,Java, SOAP

16 Integral-C—a practical environment for C programming
Graham Ross
January 1987 **ACM SIGPLAN Notices , Proceedings of the second ACM SIGSOFT/SIGPLAN software engineering symposium on Practical software development environments**, Volume 22 Issue 1
Full text available: pdf(744.42 KB) Additional Information: full citation, abstract, references, citings, index terms

Integral C1 is an industrial grade integrated programming environment for C programming on an engineering workstation. A single interactive tool replaces a syntax checking editor, a compiler, and a source-level debugger. Its multi-window user interface allows program editing and animated source level debugging, tailored to the needs of a C programmer. The compiler accepts standard C code and reacts to editing changes with function-level incremental compilation. Compilat ...

17 On the usefulness of type and liveness accuracy for garbage collection and leak detection
Martin Hirzel, Amer Diwan, Johannes Henkel
November 2002 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 24 Issue 6
Full text available: pdf(684.85 KB) Additional Information: full citation, abstract, references, index terms

The effectiveness of garbage collectors and leak detectors in identifying dead objects depends on the *accuracy* of their reachability traversal. Accuracy has two orthogonal dimensions: (i) whether the reachability traversal can distinguish between pointers and nonpointers (*type accuracy*), and (ii) whether the reachability traversal can identify memory locations that will be dereferenced in the future (*liveness accuracy*). This article presents an experimental study of the impo ...

Keywords: Conservative garbage collection, leak detection, liveness accuracy, program analysis, type accuracy

18 Technical papers: dynamic program analysis: Tracking down software bugs using automatic anomaly detection
Sudheendra Hangal, Monica S. Lam
May 2002 **Proceedings of the 24th international conference on Software engineering**
Full text available: pdf(1.30 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper introduces DIDUCE, a practical and effective tool that aids programmers in detecting complex program errors and identifying their root causes. By instrumenting a program and observing its behavior as it runs, DIDUCE dynamically formulates hypotheses of invariants obeyed by the program. DIDUCE hypothesizes the strictest invariants at the beginning, and gradually relaxes the hypothesis as violations are detected to allow for new behavior. The violations reported help users to catch soft ...

19 Just-in-time aspects: efficient dynamic weaving for Java
Andrei Popovici, Gustavo Alonso, Thomas Gross
March 2003 **Proceedings of the 2nd international conference on Aspect-oriented software development**

Full text available: pdf(1.15 MB) Additional Information: full citation, abstract, references, citings, index terms

Recent developments in service architectures suggest that run-time adaptations could be implemented with dynamic AOP. In this paper we discuss application requirements on run-time AOP support and present a system that addresses these requirements. We provide basic support for weaving using the Just-In-Time compiler, while the AOP system is treated as an exchangeable module on top of the basic support. This approach allows us to provide low run-time overhead, AOP system flexibility, and secure we ...

20 Type feedback vs. concrete type inference: a comparison of optimization techniques for object-oriented languages
Ole Agesen, Urs Hölzle
October 1995 **ACM SIGPLAN Notices , Proceedings of the tenth annual conference on Object-oriented programming systems, languages, and applications,**
Volume 30 Issue 10

Full text available: pdf(2.27 MB) Additional Information: full citation, abstract, references, citings, index terms

Two promising optimization techniques for object-oriented languages are type feedback (profile-based receiver class prediction) and concrete type inference (static analysis). We directly compare the two techniques, evaluating their effectiveness on a suite of 23 SELF programs while keeping other factors constant.Our results show that both systems inline over 95% of all sends and deliver similar overall performance with one exception: SELF's automatic coercion of machine integer ...

Results 1 - 20 of 36 Result page: **1** 2 next

CiteSeer Find: instrument AND compile AND parsing Documents Citations

Searching for **instrument and compile and parsing and stub**.
Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) CSB DBLP
No documents match Boolean query. Trying non-Boolean relevance query.
500 documents found. **Order: relevance to query.**

Daily management of an earth observation satellite.. - Lemaitre, Verfaillie (Correct)
be taken the next day from at least one of the **instruments**, with respect to the satellite trajectory ffl
ftp.cert.fr/pub/lemaitre/Papers/97-ILOG.ps

The Application Of The Supercondensed Tlm Node To.. - Trenkic.. (Correct)
media on an arbitrarily graded mesh without using **stubs** is developed. It requires less storage,
in efficiency and accuracy over conventional **stub**-loaded symmetrical condensed node (SCN) 5] used
media on an arbitrarily graded mesh without using **stubs**. This decreases computer storage requirements by
nmle.eee.nott.ac.uk/~vmt/preprints/compel95.ps.gz

The Data Reduction Expert Assistant - Miller (1992) (Correct)
only on the characteristics of the detector and **instrument** mode. The last two steps are more involved
www.stsci.edu/~miller/draco/draco-aldb.ps

Optimizing ML with Run-Time Code Generation - Leone, Lee (1995) (Correct) (112 citations)
generating code was brief, totalling 1.3 ms. **Instrument**ation revealed that the run-time code generator
We describe the design and implementation of a **compile**r that automatically translates ordinary programs
destined for a Telnet port. This packet filter must **parse** the IP header, the length of which may vary, in
foxnet.cs.cmu.edu/~petel/papers/staged/mleone-pldi96.ps

Mechanisms and Interfaces for Software-Extended Coherent Shared.. - Chaiken (1994) (Correct) (4 citations)
are very different, however. Mtool **instruments** code on the processor side and presents
about memory usage from the runtime system to the **compile**r. The **compile**r uses this information to
ftp.cag.lcs.mit.edu/pub/papers/chaiken-dissert-1-10.ps.Z

Segregatory Coordination and Ellipsis in Text Generation - Shaw (1998) (Correct) (6 citations)
will not address common problems associated with **parsing**, such as disambiguation and construction of
www.cs.columbia.edu/~shaw/papers/colingacl98.ps.gz

Intelligent Computing About Complex Dynamical Systems - Zhao (1994) (Correct)
engineering simulation arise from the lack of (1) **pars**imonious representations capturing the essence of
www.cis.ohio-state.edu/insight/papers/mcs.ps

Optimized Software Synthesis for Digital Signal.. - Jürgen Teich.. (1998) (Correct) (3 citations)
and consumed by each actor is fixed and known at **compile**-time. Example 1.1 Figure 1.1 shows a simple SDF
ftp.tik.ee.ethz.ch/pub/people/zitzler/TZB1998a.ps.gz

Natural Deduction for Intuitionistic Non-Commutative Linear.. - Polakow, Pfenning (1999) (Correct) (15 citations)
and applied it to algorithms for natural language **parsing**, sorting, and execution of abstract machines
pecan.srv.cs.cmu.edu/afs/cs.cmu.edu/user/fp/www/papers/tlca99.ps.gz

Learning Planning Operators by Observation and Practice - Wang (1994) (Correct) (15 citations)
www.rpal.rockwell.com/~mei/aips94.ps

Formalising Abilities and Opportunities of Agents - van Linder, van der Hoek, Meyer (1998) (Correct) (7 citations)
ftp.cs.uu.nl/pub/RUU/CS/techreps/CS-1998/1998-08.ps.gz

Development, Learning and Evolution in Animats - Kodjabachian, Meyer (1994) (Correct) (2 citations)
www.biologie.ens.fr/fr/animatlab/perso/kodjaba/jkjamperac.ps.gz

An Object Calculus with Algebraic Rewriting - Compagnoni, Fernández (Correct)
fields. In a language based on our calculus the **compile**r can make the translation from the algebraic

www.ens.fr/~maribel/papers/PLILP97.ps.gz

Statistical Learning, Localization, and Identification of.. - Hornegger, Niemann (1995) (Correct) (1 citation)
www5.informatik.uni-erlangen.de/TeX/Literatur/ps-dir/1995/Hornegger95:SLL.ps.gz

Fourth And Fifth Order Efficiency: Fisher Information - Kano (Correct)
koko15.hus.osaka-u.ac.jp/members/kano/research/./dvi/fisher.ps

Frames, Objects and Relations: Three Semantic.. - Norrie, Reimer.. (1994) (Correct)
in COCOON. To increase the possibilities for **compile**time optimisation, we designed the mapping such
www.globis.ethz.ch/publications/docs/1994d-nrlrs-krdb.ps.gz

Towards 3-D model-based tracking and recognition of human.. - Gavrila, Davis (1995) (Correct)
the pose parameter space. Finally, we plan to **compile** a large data set of "everyday" activities to
www.umiacs.umd.edu/users/gavrila/iwafgr.ps.Z

Uniform Reconstruction of Gaussian Processes - Müller-Gronbach, Ritter (1995) (Correct) (1 citation)
ftp.math.fu-berlin.de/pub/math/publ/pre/1995/pr-a-95-26.ps.Z

Topic Detection and Tracking Pilot Study - Allan, Carbonell, Doddington.. (1998) (Correct) (3 citations)
a threshold. A second approach used a shallow **parser** to extract nouns and noun phrases (rather than
www.cs.cmu.edu/~yiming/papers.yy/tdt1-final-report.ps

The Alloc Stream Facility: A Redesign of Application-Level .. - Krieger, Stumm, Unrau (1994) (Correct) (13 citations)
5 With a macro, the code is inserted by a **compile**r pre processor wherever the macro is called.
ftp.cs.toronto.edu/pub/parallel/Krieger_etal_IEEEComp94.ps.Z

First 20 documents Next 20

Try your query at: Google (CiteSeer) Google (Web) CSB DBLP

IEEE Xplore®
RELEASE 1.8

Welcome
United States Patent and Trademark Office

11
1
1

Help FAQ Terms IEEE Peer Review Quick Links

» Sea

Welcome to IEEE Xplore®

- ○ Home
- ○ What Can I Access?
- ○ Log-out

Tables of Contents

- ○ Journals & Magazines
- ○ Conference Proceedings
- ○ Standards

Search

- ○ By Author
- ○ Basic
- ○ Advanced

Member Services

- ○ Join IEEE
- ○ Establish IEEE Web Account
- ○ Access the IEEE Member Digital Library

IEEE Enterprise

- ○ Access the IEEE Enterprise File Cabinet

Print Format

Your search matched **11** of **1069805** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri new one in the text box.

instrument<and>stub Search

□ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 **Noise measurements of microwave transistors using an uncalibrate mechanical stub tuner and a built-in reverse six-port reflectometer**
Di-Luan Le; Ghannouchi, F.M.;
Instrumentation and Measurement, IEEE Transactions on , Volume: 44 , Issue 4 , Aug. 1995
Pages:847 - 852

[Abstract] [PDF Full-Text (396 KB)] **IEEE JNL**

2 **Modeling, simulation, and measurement considerations of high-spee digital buses**
Novak, I.;
Instrumentation and Measurement, IEEE Transactions on , Volume: 41 , Issue 6 , Dec. 1992
Pages:921 - 925

[Abstract] [PDF Full-Text (376 KB)] **IEEE JNL**

3 **Modeling, simulation, and measurement considerations of high-spee digital buses**
Novak, I.;
Instrumentation and Measurement Technology Conference, 1992. IMTC '92., IEEE , 12-14 May 1992
Pages:147 - 151

[Abstract] [PDF Full-Text (372 KB)] **IEEE CNF**

4 **The Level-2 muon trigger at D0**
Fortner, M.; Maciel, A.; Evans, H.; Kothari, B.; Uzunyan, S.;
Nuclear Science, IEEE Transactions on , Volume: 49 , Issue: 4 , Aug. 2002
Pages:1589 - 1592

[Abstract] [PDF Full-Text (197 KB)] **IEEE JNL**

5 **Measurement and simulation of crosstalk reduction by discrete discontinuities along coupled PCB traces**
Novak, I.; Eged, B.; Hatvani, L.;
Instrumentation and Measurement, IEEE Transactions on , Volume: 43 , Issue 2 , Apr 1994
Pages:170 - 175

[Abstract] [PDF Full-Text (400 KB)] **IEEE JNL**

6 **The level-2 muon trigger at D0**
Fortner, M.; Maciel, A.; Evans, H.; Kothari, B.; Uzunyan, S.;
Nuclear Science Symposium Conference Record, 2000 IEEE , Volume: 2 , 15-Oct. 2000
Pages:12/6 - 12/9 vol.2

[Abstract] [PDF Full-Text (344 KB)] **IEEE CNF**

7 **Reduction and characterization of a drift error in measurement of sm antennas using a network analyzer**
Ida, I.; Sato, J.; Sekizawa, T.; Yoshimura, H.; Ito, K.;
Antennas, Propagation and EM Theory, 2000. Proceedings. ISAPE 2000. 5th International Symposium on , 15-18 Aug. 2000
Pages:504 - 507

[Abstract] [PDF Full-Text (288 KB)] **IEEE CNF**

8 **Coupling cavity damper for the ARES**
Naito, F.; Akai, K.; Akasaka, N.; Ezura, E.; Kageyama, T.; Mizuno, H.; Nakan H.; Takeuchi, Y.; Yamazaki, Y.; Kobayashi, T.;
Particle Accelerator Conference, 1997. Proceedings of the 1997 , Volume: 3 , May 1997
Pages:2977 - 2979 vol.3

[Abstract] [PDF Full-Text (244 KB)] **IEEE CNF**

9 **A design pattern based approach to generating synchronization ada from annotated IDL**
Jacobsen, H.-A.; Kramer, B.J.;
Automated Software Engineering, 1998. Proceedings. 13th IEEE International Conference on , 13-16 Oct. 1998
Pages:63 - 72

[Abstract] [PDF Full-Text (236 KB)] **IEEE CNF**

10 **System control and data acquisition of the two new FWCD RF syste at DIII-D**
Harris, T.E.; Allen, J.C.; Cary, W.P.; Ferguson, S.W.; Petty, C.C.; Pinsker, R.I
Fusion Engineering, 1995. SOFE '95. 'Seeking a New Energy Era'., 16th IEEE/ Symposium , Volume: 2 , 30 Sept.-5 Oct. 1995
Pages:878 - 880 vol.2

[Abstract] [PDF Full-Text (320 KB)] **IEEE CNF**

11 **The ICRF computer control and monitoring system**
Mervine, J.; Sichta, P.; Wilson, R.; Giles, D.; Skelly, G.;

Fusion Engineering, 1989. Proceedings., IEEE Thirteenth Symposium on , 2-6 1989
Pages:455 - 458 vol.1

[Abstract] [PDF Full-Text (184 KB)] **IEEE CNF**

Home | Log-out | Journals | Conference Proceedings | Standards | Search by Author | Basic Search | Advanced Search | Join IEEE | Web Account | New this week | OPAC Linking Information | Your Feedback | Technical Support | Email Alerting | No Robots Please | Release Notes | IEEE Online Publications | Help | FAQ| Terms | Back to Top